*Copy the projects into \examples\dfu folder in SDK v14.2 to compile. Four SPI pins needed: SCK, MISO, MOSI, SS. Configure them in sdk\_config.h. By default they are pins P0.22 23 24 25 respectively*

# SPIS DFU bootloader:

The SPIS bootloader follow exactly what inside the UART bootloader\_secure\_serial. Only difference is SPI transport layer is used instead of UART . And because of SPI used, we don’t need to use SLIP encapsulation to send a packet. So the protocol is the same as in the documentation for UART [here](http://infocenter.nordicsemi.com/topic/com.nordic.infocenter.sdk5.v14.2.0/lib_dfu_transport_serial.html?cp=4_0_0_3_5_2_3) except for SLIP.

# SPI DFU Master:

The DFUMaster\_SPI application is set up to allow an application image, a SoftDevice and/or bootloader image to be stored in the flash of a 512kB nRF52832 device. The host application itself will run from 0x0000. There are dedicated locations for the signed init packets as well.

The flash configuration should be as follows:

|  |  |
| --- | --- |
| 0x31000 | Softdevice and/or bootloader image |
| 0x30000 | Init packet for bootloader and/or softdevice image |
| 0x10000 | Application image |
| 0x9000 | Init packet for application image |
| 0x0000 | DFU master application |

# How to test

The host application itself can be programmed as any other application.

1. Compile the bootloader with the public key you generated. Flash the target board with softdevice and the bootloader. Flash the DFU master application on the master board.
2. Generate the DFU .zip package file as you do on normal DFU (make sure you use private key as used to generate the public key)
3. Open the .zip file and unzip the init .dat file and the binary image .bin of the application or the softdevice/bootloader.
4. Rename the init .dat file to a .bin file (make sure you don’t overwrite the image .bin file) e.g init\_packet.bin
5. Use J-Link commander to flash the binary file to the right part of the flash in the host
   1. Run the command line in the folder where you have your binary files
   2. From the command line, start J-Link commander by running ***jlink*** (this requires the *C:\Program Files (x86)\SEGGER\JLink\_VXXX* folder to be in your PATH environment settings in Windows)
   3. In J-Link commander, type the following commands, this is for writing application, the similar applied for softdevice/bootloader just choose the right location:
      1. ***connect***
      2. ***<Select nRF52832 and select SWD interface>***
      3. ***loadbin init\_packet.bin 0x9000***
      4. ***loadbin app\_image.bin 0x10000***
      5. ***r***
      6. ***g***
6. When the DFU Master started it will send a SPI ping command. After that you can choose to update application or softdevice/bootloader by pressing Button 1 or button 2 on the DFU master board.
7. LED 1 will lid when the image is being transferred

**Limitation**: As it’s just a test DFU master, there is no status check, no MTU request, no packets notification enable, no continual if the link is broken (it will start from scratch). The response code from SPI peer is captured by polling every 1 ms.